

char *c;

int *i;

float *f;

char text[40];

int num[100];

char ch;

~~X num = 100, 200, 300;~~

~~X text = "message";~~

~~c = ch;~~

→ text[0] = 'c';

ch = text[14];

num[11] = 10;

printf("%d\n", num[0]);

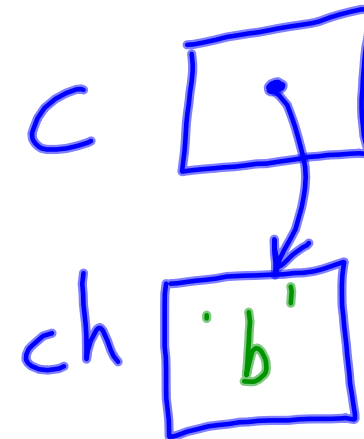
c = &ch;

```
int *c, ch;
```

```
c = &ch;
```

```
ch = 'a';
```

```
*c = 'b';
```



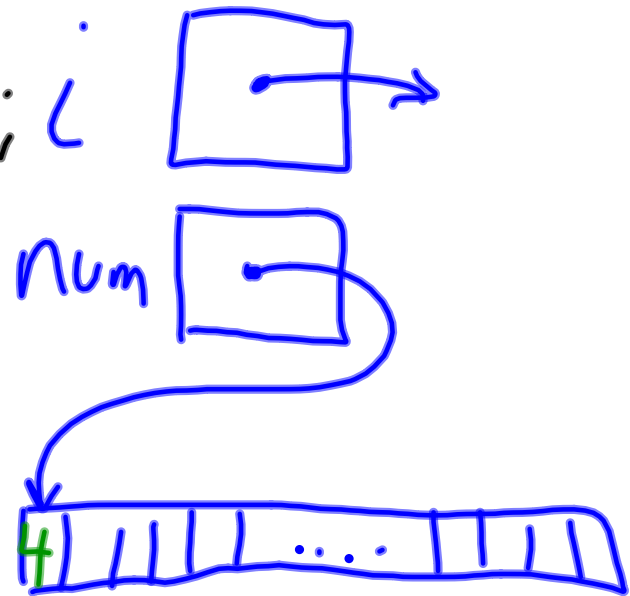
```
int *i, num[100];
```

`i = num;` When declaring an array variable
1. declare an `int *`, named `num`.
2. allocate space for 100 ints.

`num[0] = 4;` 3. point `num` at the allocated memory

```
printf("%d %d\n", num[0], *i);
```

`*num, i[0]` `num`



```
void print_nums ( int *n ) {  
    while ( *n != -1 ) {  
        printf("%d\n", *n);  
        n++;  
    }  
}
```

```
print_nums(num);
```

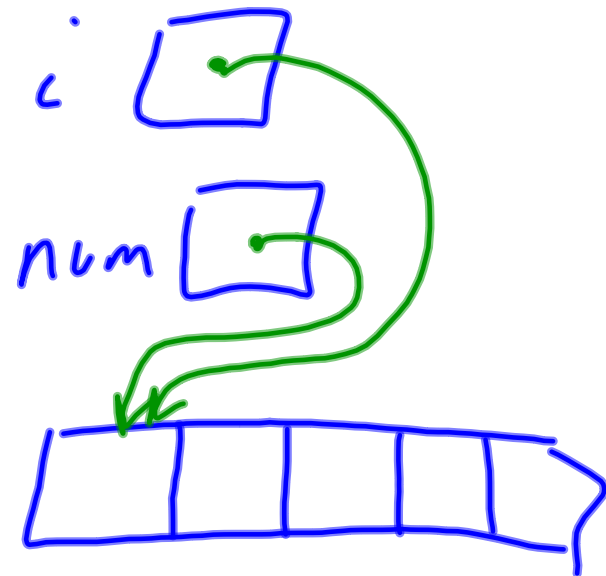
```
int *i, num[10];
```

```
i = num;
```

```
int value = num[0];  
          = *i;
```

```
value = num[1];  
       = *(i + 1);
```

num → 0x2341Acd



```
struct f {  
    int a;  
    char b;  
};
```

```
struct f *p, v;  
p = &v;
```

```
typedef struct {  
    int i;  
    char x;  
} S;
```

```
S *r, w;  
r = &w;
```